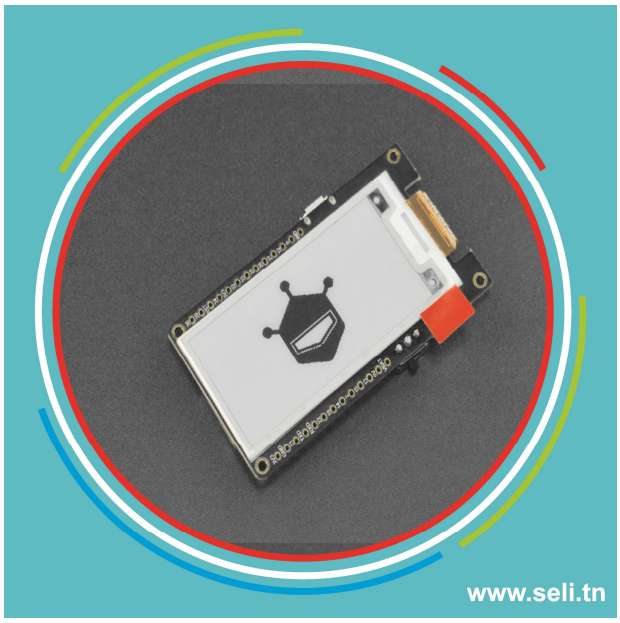
AFFICHEURE-LINK 2.13 POUCES POUR ESP32 DFR0676



Introduction

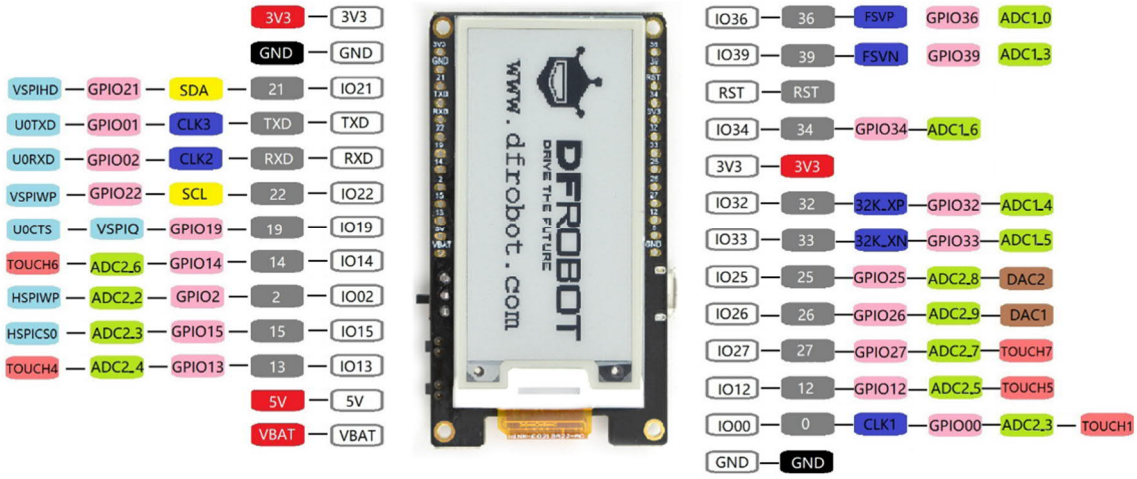
This is an ESP32-integrated 2.13”e-ink display with built-in WiFi and Bluetooth functions that can connect to Alibaba Cloud to achieve remote display control. It supports controlling other devices via its IO ports, which greatly saves development cost. This e -ink display module can clearly display Chinese characters, pattern, English in black and white. When powered off, the display will continue to display the last screen. In addition, it features low power consumption and a wide view angle.

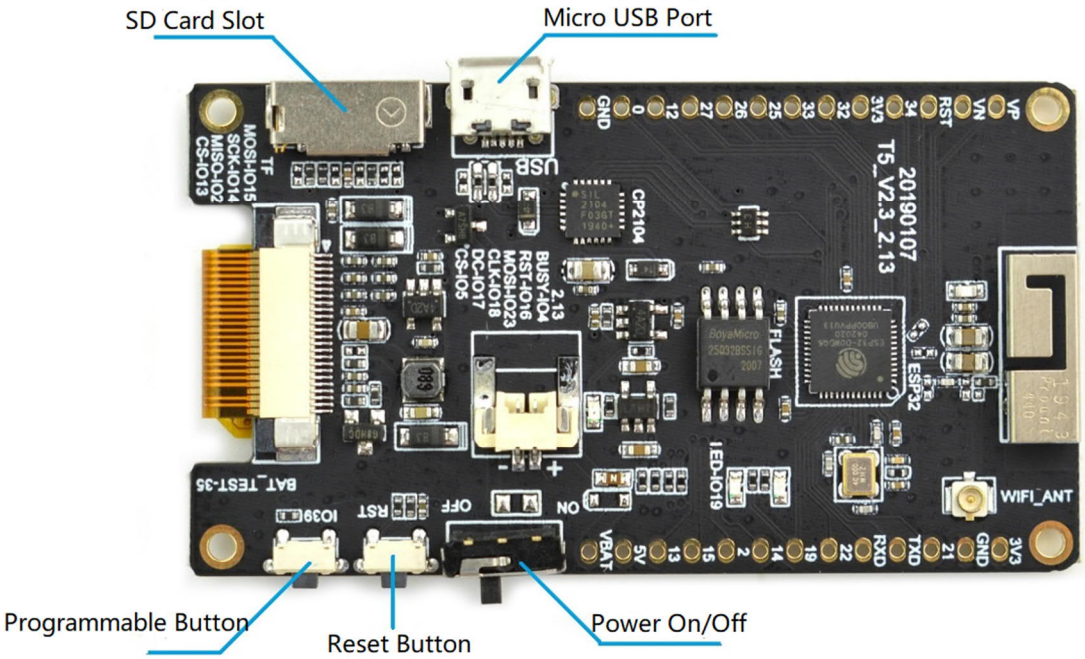
With a resolution of 250*122, this product is very suitable for displaying static pattern or text, for instance, price tag, name badge, message board, etc. When connected to WiFi, it can be turned into a calendar, electronic clock, weather forecast display, and so on.

Specification

- Operating Voltage: 3.3V or 5V
- Display Color: black, white
- Grey Level: 2
- Refresh Time: 2 seconds
- Driver Chip: GDEH0213B72
- Resolution: 250*122
- Display Area: 48.55*23.17mm

Board Overview





Tutorial

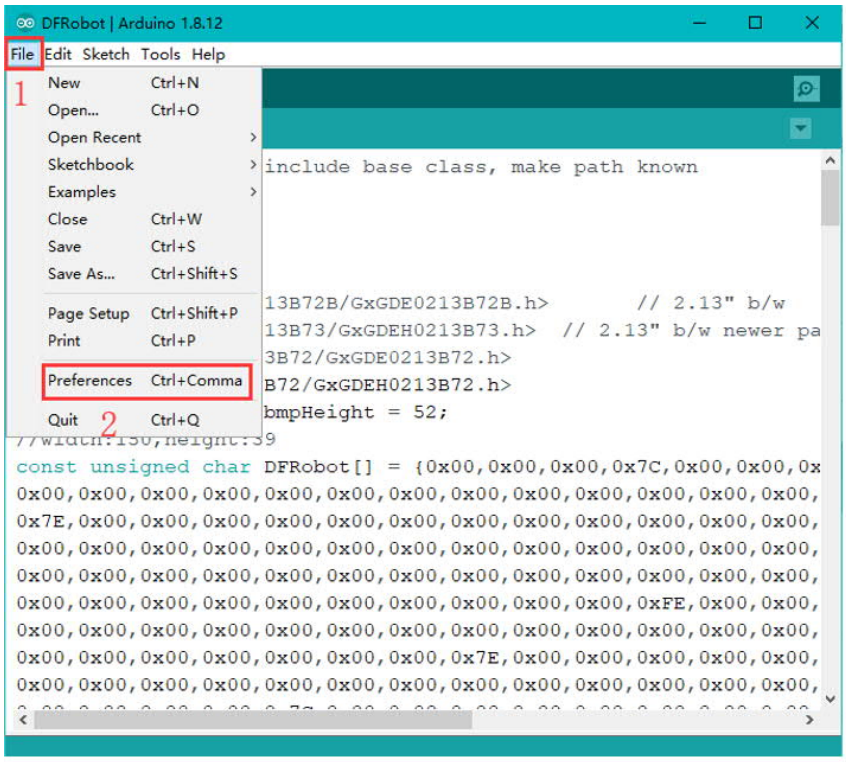
Requirements

- Hardware
  - ESP32 e-Ink Display x 1
  - Micro USB x 1

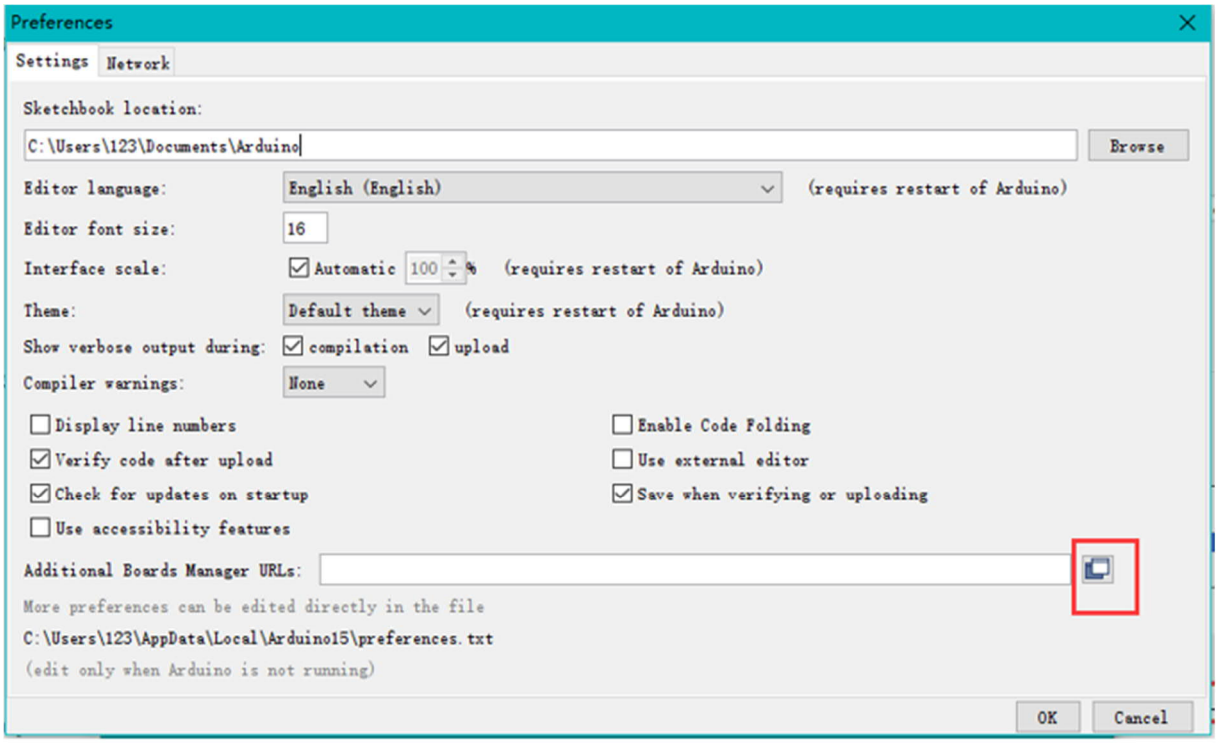
- Software
    - Arduino IDE
    - Download and install the GxEPD Library (About how to install the library?)
    - Click to download the Adafruit-GFX-Library .

Install Development Board Driver

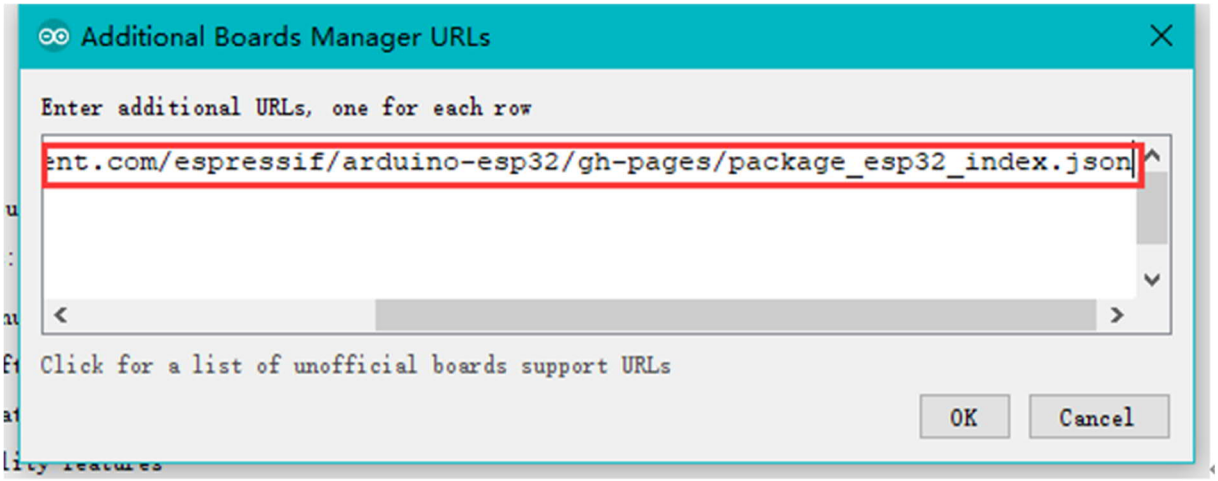
1. Open Arduino IDE, add the download address of the development driver. Select File->Preferences in the menu(or use keyboard shortcut).



2. Click the icon at the right of the Additional Boards Manager URLs.

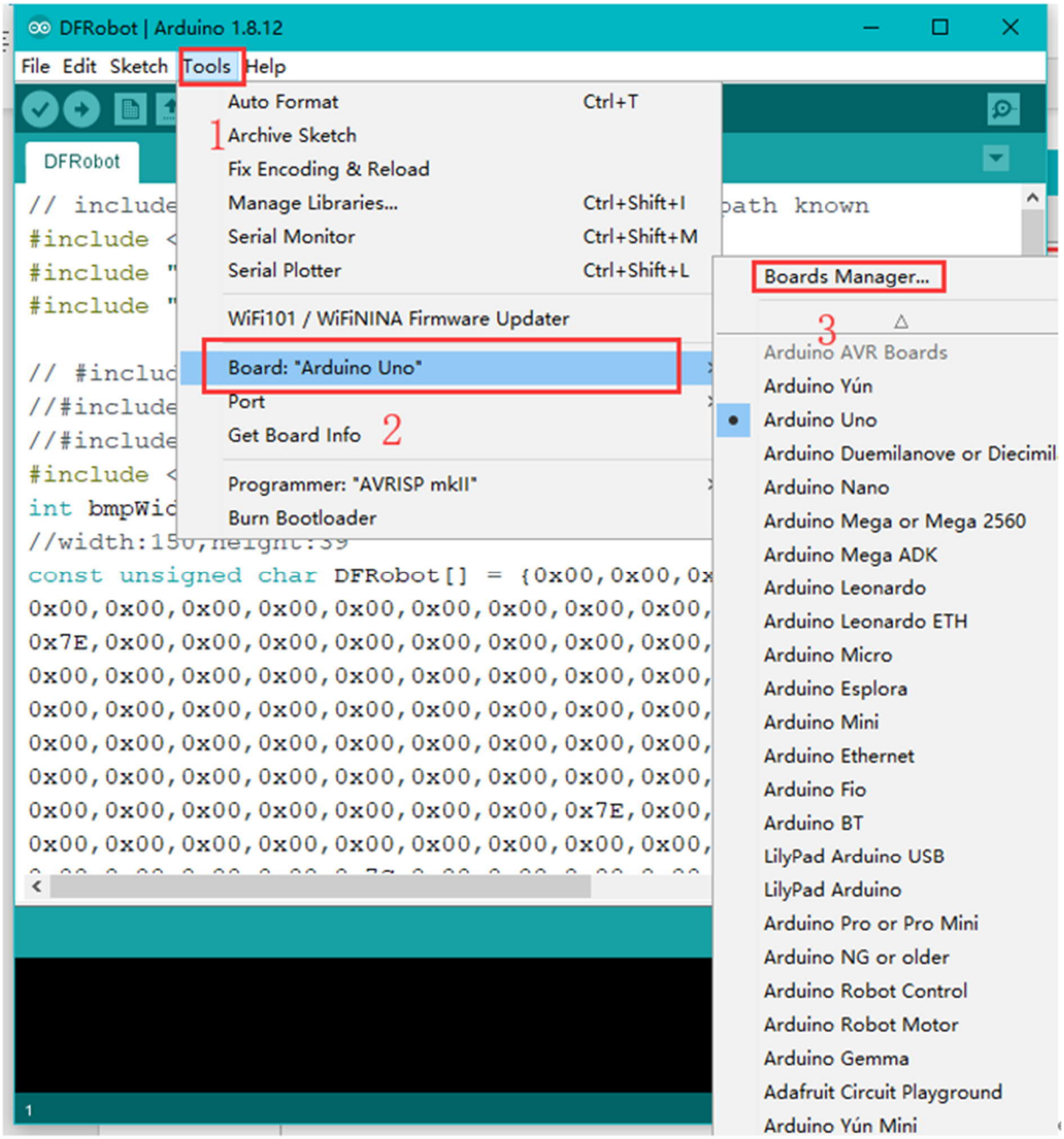


3. Copy the URL address and paste it to the last line in the pop-up box: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

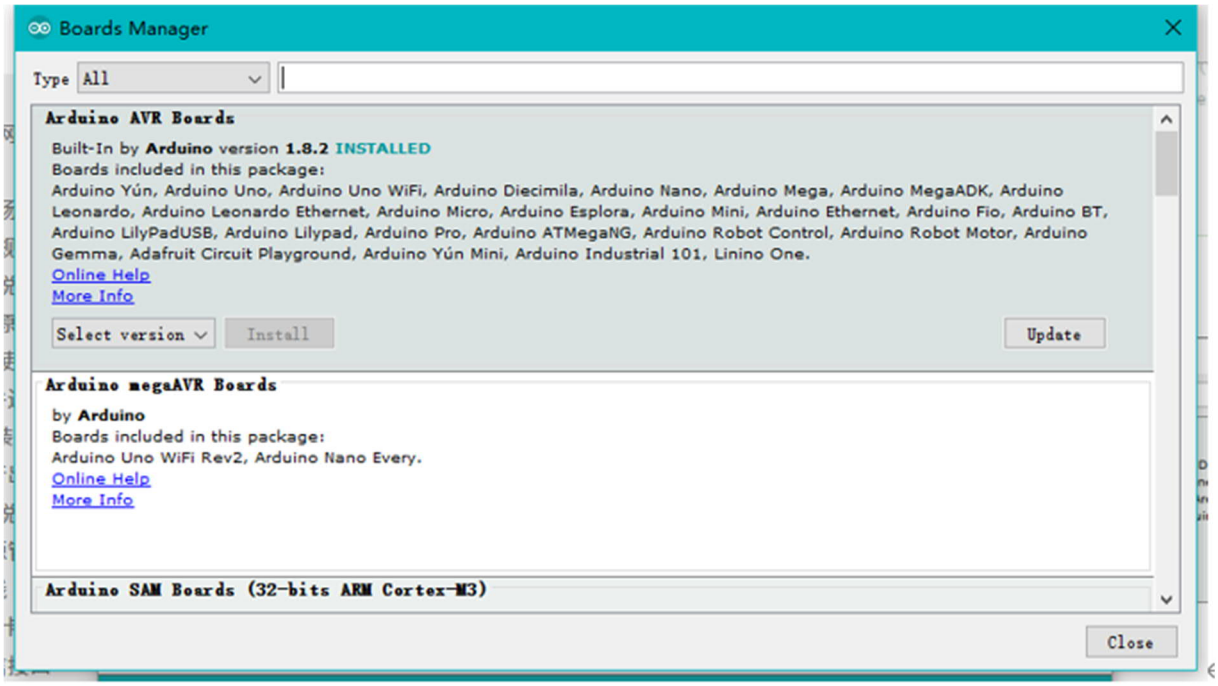


4. Click "OK" in the Additional Boards Manager URLs and Preferences.

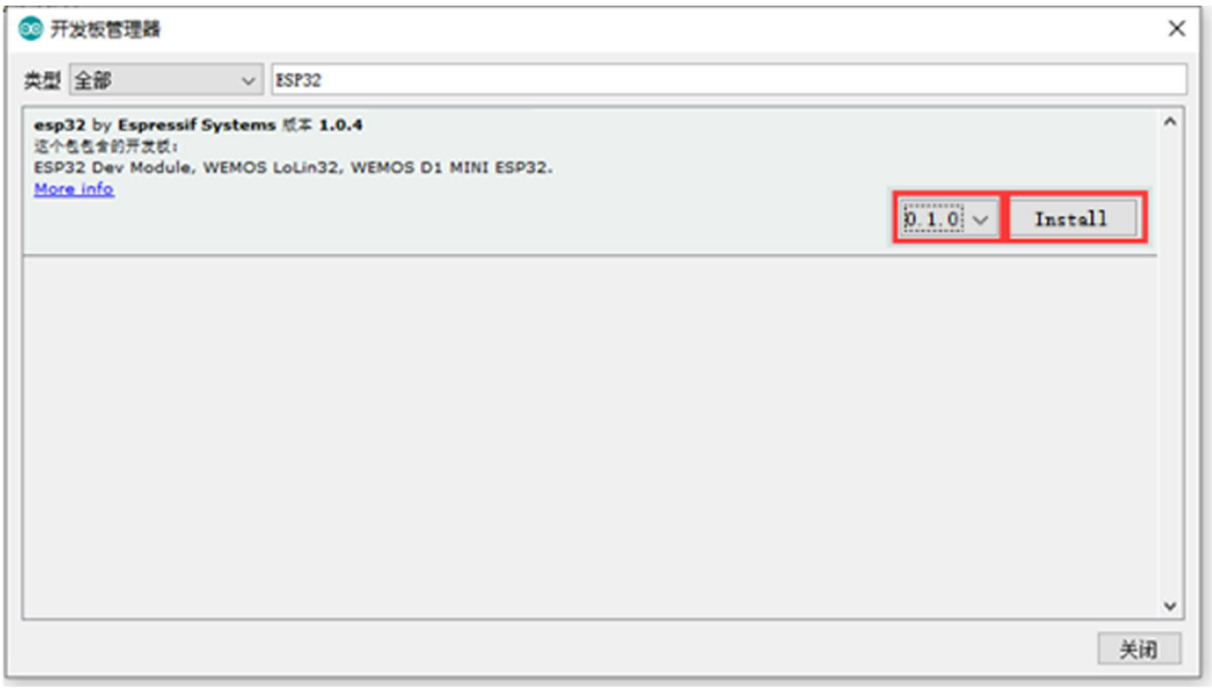
5. Select Tools->Board->Boards Manager in the menu.



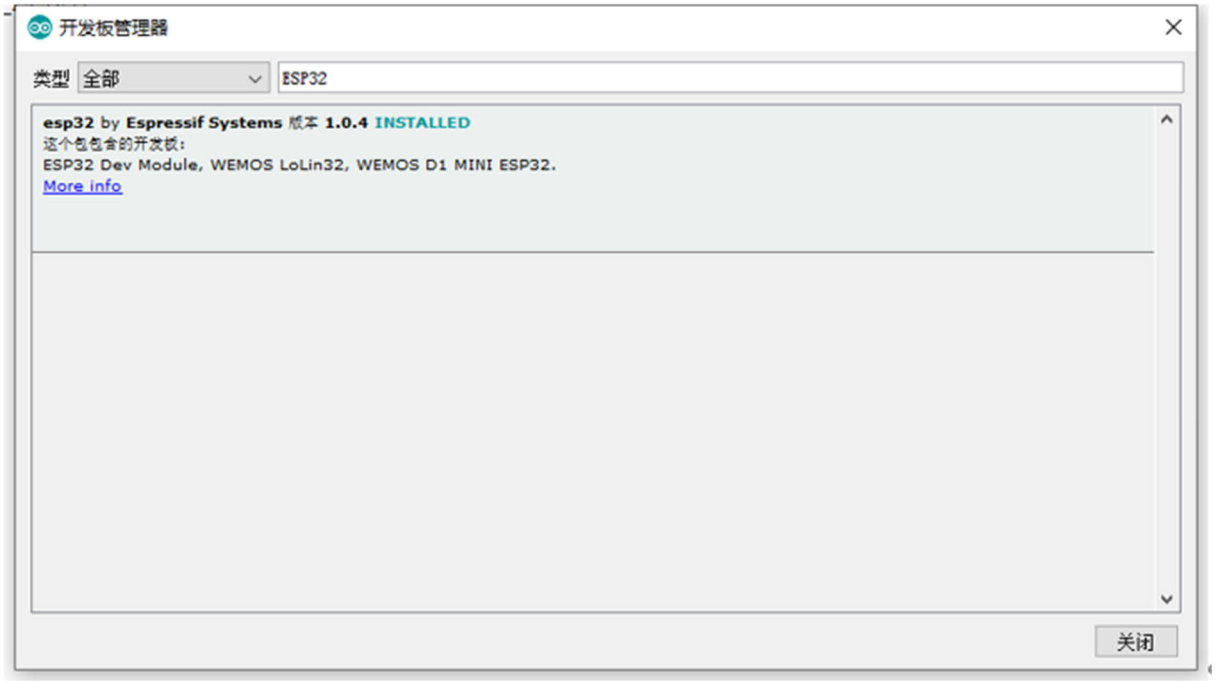
6. The boards Manager will pop up, and the board information will be updated automatically.



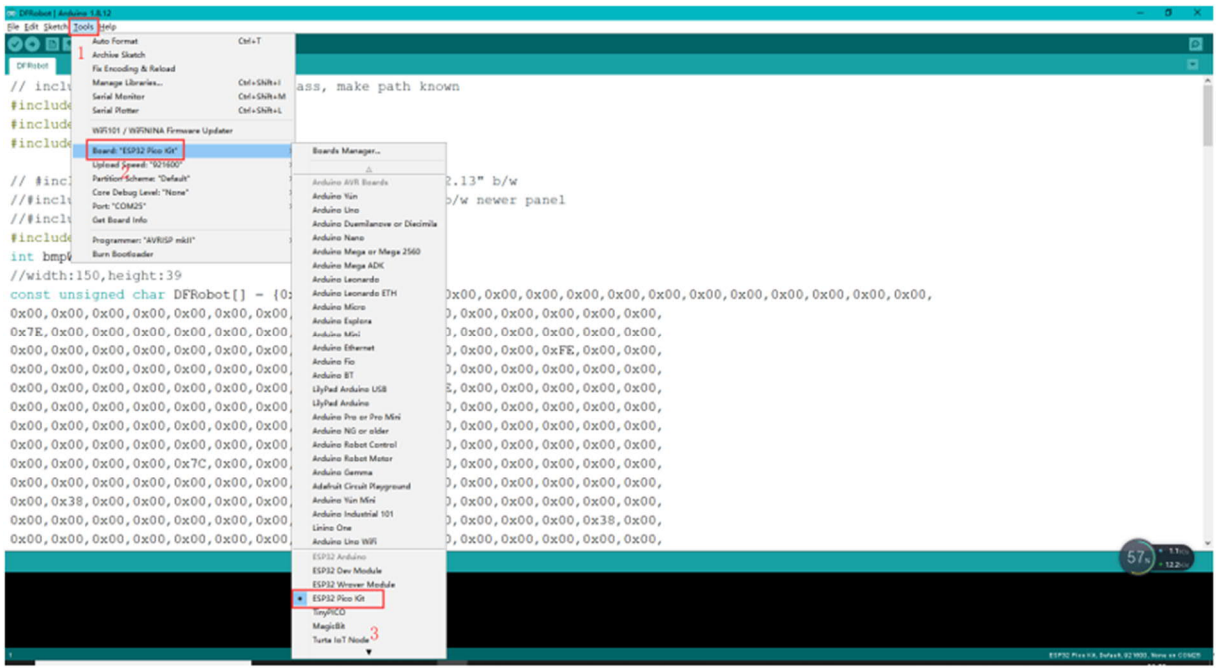
7. Search "ESP32" when the update completed. Select the latest version of development board, and click "Install" to install the driver.



8. Installation done!



9. Select Tools->Board->ESP32 Pico Kit, now you can burn programs into the board or check the serail printed data.



Example 1: display image and characters

```
// include library, include base class, make path known
#include <GxEPD.h>
#include "SD.h"
#include "SPI.h"
```

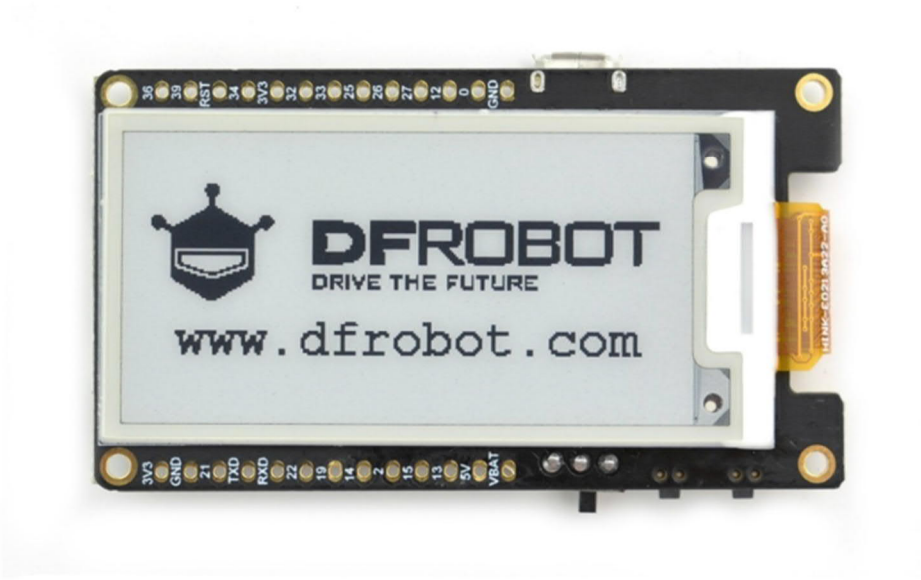
```
//Since there are multiple versions of the screen, if there is a flower screen
after downloading the program, please test the following four header files again!
#include <GxDEPG0213BN/GxDEPG0213BN.h>
//#include <GxGDE0213B1/GxGDE0213B1.h>      // 2.13" b/w
//#include <GxGDEH0213B72/GxGDEH0213B72.h>  // 2.13" b/w new panel
//#include <GxGDEH0213B73/GxGDEH0213B73.h>  // 2.13" b/w newer panel
int bmpWidth = 232, bmpHeight = 52;
//width:150,height:39
const unsigned char DFRobot[] =
{0x00,0x00,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x7E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xFF,
0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xFF,0xE0,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x78,0x00,0x3F,0xFF,0xF8,0x00,0x3C,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0xFC,0x00,0x7F,0xFF,0xFE,0x00,0x7E,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0xFE,0x01,0xFF,0xFF,0xFF,0x00,0x7F,0x00,0x00,0x7F,0xFE,0x00,0x1F,0xFF,0xFE,0x7F,
0xFF,0x80,0x3F,0xFF,0x03,0xFF,0xFC,0x01,0xFF,0xF8,0x3F,0xFF,0xFC,0xFE,0x07,0xFF,
0xFF,0xFF,0xC0,0xFF,0x00,0x00,0x7F,0xFF,0xF0,0x1F,0xFF,0xFF,0x7F,0xFF,0xF1,0xFF,
0xFF,0xE3,0xFF,0xFF,0x0F,0xFF,0xFF,0x7F,0xFF,0xFC,0xFF,0x9F,0xFF,0xFF,0xFF,0xF3,
0xFE,0x00,0x00,0x7F,0xFF,0xFC,0x1F,0xFF,0xFF,0x7F,0xFF,0xF1,0xFF,0xFF,0xF3,0xFF,
0xFF,0x9F,0xFF,0xFF,0x7F,0xFF,0xFC,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFC,0x00,0x00,
0x7F,0xFF,0xFE,0x1F,0xFF,0xFF,0x7F,0xFF,0xF3,0xFF,0xFF,0xF3,0xFF,0xFF,0x9F,0xFF,
0xFF,0x7F,0xFF,0xFC,0x33,0xFF,0xFF,0xFF,0xFF,0xFF,0x98,0x00,0x00,0x7F,0xFF,0xFF,
0x1F,0xFF,0xFF,0x78,0x00,0xFB,0xC0,0x00,0xF3,0xC0,0x07,0x9E,0x00,0x07,0x80,0x38,
0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x7F,0xFF,0xFF,0x1F,0xFF,0xFF,
0x78,0x00,0xFB,0xC0,0x00,0xF3,0xC0,0x07,0x9E,0x00,0x07,0x80,0x3C,0x00,0x00,0x7F,
0xFF,0xFF,0xFF,0xFE,0x00,0x00,0x00,0x7E,0x00,0x7F,0x9F,0x80,0x00,0x78,0x00,0xFB,
0xC0,0x00,0xF3,0xC0,0x07,0x9E,0x00,0x07,0x80,0x3C,0x00,0x00,0x7F,0xFF,0xFF,0xFF,
0xFE,0x00,0x00,0x00,0x7F,0x00,0x3F,0x9F,0x80,0x00,0x78,0x00,0xFB,0xC0,0x00,0xF3,
0xC0,0x07,0x9E,0x00,0x07,0x80,0x3C,0x00,0x00,0x7F,0xFF,0xFF,0xFF,0xFE,0x00,0x00,
0x00,0x7F,0x00,0x1F,0x9F,0xFF,0xF8,0x78,0x00,0xFB,0xC0,0x00,0xF3,0xFF,0xFF,0x9E,
0x00,0x07,0x80,0x3C,0x00,0x00,0x7F,0xFF,0xFF,0xFF,0xFE,0x00,0x00,0x00,0x7F,0x00,
0x1F,0x9F,0xFF,0xF8,0x7F,0xFF,0xF3,0xC0,0x00,0xF3,0xFF,0xFE,0x1E,0x00,0x07,0x80,
0x3C,0x00,0x00,0x7F,0xFF,0xFF,0xFF,0xFE,0x00,0x00,0x00,0x7F,0x00,0x1F,0x9F,0xFF,
0xF8,0x7F,0xFF,0xF3,0xC0,0x00,0xF3,0xFF,0xFF,0x1E,0x00,0x07,0x80,0x3C,0x00,0x00,
0x7F,0xFF,0xFF,0xFF,0xFE,0x00,0x00,0x00,0x7F,0x00,0x3F,0x9F,0xFF,0xF8,0x7F,0xFF,
```

```
0xE3,0xC0,0x00,0xF3,0xC0,0x07,0x9E,0x00,0x07,0x80,0x3C,0x00,0x00,0x7F,0xFF,0xFF,
0xFF,0xFE,0x00,0x00,0x00,0x7F,0x00,0x3F,0x9F,0xFF,0xF8,0x78,0x3E,0x03,0xC0,0x00,
0xF3,0xC0,0x07,0x9E,0x00,0x07,0x80,0x3C,0x00,0x00,0x60,0x00,0x00,0x00,0x0E,0x00,
0x00,0x00,0x7E,0x00,0xFF,0x9F,0x80,0x00,0x78,0x1F,0x03,0xC0,0x00,0xF3,0xC0,0x07,
0x9E,0x00,0x07,0x80,0x3C,0x00,0x00,0x6F,0xFF,0xFF,0xFF,0xEE,0x00,0x00,0x00,0x7F,
0xFF,0xFF,0x1F,0xC0,0x00,0x78,0x0F,0x83,0xC0,0x00,0xF3,0xC0,0x07,0x9E,0x00,0x07,
0x80,0x3C,0x00,0x00,0x6F,0xFF,0xFF,0xFF,0xEE,0x00,0x00,0x00,0x7F,0xFF,0xFF,0x1F,
0xC0,0x00,0x78,0x07,0xC3,0xC0,0x00,0xF3,0xC0,0x07,0x9E,0x00,0x07,0x80,0x3C,0x00,
0x00,0x6C,0x00,0x00,0x00,0x6E,0x00,0x00,0x00,0x7F,0xFF,0xFE,0x1F,0xC0,0x00,0x78,
0x03,0xE3,0xFF,0xFF,0xF3,0xFF,0xFF,0x9F,0xFF,0xFF,0x00,0x3C,0x00,0x00,0x6C,0x00,
0x00,0x00,0x6E,0x00,0x00,0x00,0x7F,0xFF,0xFC,0x1F,0xC0,0x00,0x78,0x01,0xF9,0xFF,
0xFF,0xF3,0xFF,0xFF,0x9F,0xFF,0xFF,0x00,0x3C,0x00,0x00,0x6C,0x00,0x00,0x00,0x6E,
0x00,0x00,0x00,0x7F,0xFF,0xF0,0x1F,0xC0,0x00,0x78,0x00,0xFD,0xFF,0xFF,0xE3,0xFF,
0xFF,0x0F,0xFF,0xFF,0x00,0x3C,0x00,0x00,0x6E,0x00,0x00,0x00,0xEE,0x00,0x00,0x00,
0x7F,0xFE,0x00,0x1F,0xC0,0x00,0x78,0x00,0x7C,0x3F,0xFF,0x03,0xFF,0xF8,0x03,0xFF,
0xF8,0x00,0x38,0x00,0x00,0x67,0xC0,0x00,0x07,0xCE,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x70,0xF8,0x00,0x3E,0x1E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7E,
0x1F,0x01,0xF0,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xC3,0xFF,0x87,
0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xF8,0x3C,0x3F,0xFE,0x00,0x00,
0x00,0x7E,0x7E,0xD0,0xBF,0x0F,0xD8,0x9F,0x83,0xEC,0x5F,0xB1,0x7F,0x7E,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xFF,0x81,0xFF,0xFC,0x00,0x00,0x00,0x7F,0x7E,
0xD9,0xBF,0x0F,0xD8,0x9F,0x83,0xEC,0x5F,0xB1,0x7F,0x7E,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x3F,0xFF,0xFF,0xFF,0xFC,0x00,0x00,0x00,0x63,0x62,0xD9,0xA0,0x03,
0x1F,0x90,0x02,0x0C,0x46,0x31,0x63,0x40,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x1F,0xFF,0xFF,0xFF,0xF0,0x00,0x00,0x00,0x63,0x7E,0xCF,0x3E,0x03,0x1F,0x9F,0x03,
0xCC,0x46,0x31,0x7F,0x78,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0xFF,0xFF,
0xFF,0xC0,0x00,0x00,0x00,0x63,0x6C,0xCF,0x30,0x03,0x18,0x98,0x03,0xCC,0x46,0x21,
0x6E,0x40,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0xFF,0xFF,0xFF,0x80,0x00,
0x00,0x00,0x7F,0x66,0xC6,0x3F,0x03,0x18,0x9F,0x83,0x0F,0xC6,0x3F,0x63,0x7E,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFE,0x00,0x00,0x00,0x00,0x7C,
0x62,0xC6,0x3F,0x03,0x18,0x9F,0x82,0x07,0xC6,0x1F,0x61,0x7E,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x3F,0xFF,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x0F,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,
0xFF,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0xFF,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00};
// FreeFonts from Adafruit_GFX
#include <Fonts/FreeMonoBold9pt7b.h>
#include <Fonts/FreeMonoBold12pt7b.h>
#include <Fonts/FreeMonoBold18pt7b.h>
#include <Fonts/FreeMonoBold24pt7b.h>


#include <GxIO/GxIO_SPI/GxIO_SPI.h>
#include <GxIO/GxIO.h>

#define SPI_MOSI 23
#define SPI_MISO -1
#define SPI_CLK 18
```

```
#define ELINK_SS 5
#define ELINK_BUSY 4
#define ELINK_RESET 16
#define ELINK_DC 17

#define SDCARD_SS 13
#define SDCARD_CLK 14
#define SDCARD_MOSI 15
#define SDCARD_MISO 2

#define BUTTON_PIN 39


GxIO_Class io(SPI, /*CS=5*/ ELINK_SS, /*DC=*/ ELINK_DC, /*RST=*/ ELINK_RESET);
GxEPD_Class display(io, /*RST=*/ ELINK_RESET, /*BUSY=*/ ELINK_BUSY);

SPIClass sdSPI(VSPI);


const char *Website = "www.dfrobot.com";
bool sdOK = false;
int startX = 9, startY = 20;

void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.println("setup");
  SPI.begin(SPI_CLK, SPI_MISO, SPI_MOSI, ELINK_SS);
  display.init(); // enable diagnostic output on Serial

  display.setRotation(1);
  display.fillScreen(GxEPD_WHITE);
  display.setTextColor(GxEPD_BLACK);
  display.setFont(&FreeMonoBold12pt7b);
  display.setCursor(0, 0);

  sdSPI.begin(SDCARD_CLK, SDCARD_MISO, SDCARD_MOSI, SDCARD_SS);

  if (!SD.begin(SDCARD_SS, sdSPI)){
  sdOK = false;
  } else {
  sdOK = true;
  }

  display.fillScreen(GxEPD_WHITE);

  display.drawBitmap(DFRobot, startX, startY,  bmpWidth, bmpHeight, GxEPD_BLACK);

  display.setCursor(20,95);

  display.println(Website);

  display.setTextColor(GxEPD_BLACK);

  display.update();

  // goto sleep
  esp_sleep_enable_ext0_wakeup((gpio_num_t)BUTTON_PIN, LOW);
```

```
  esp_deep_sleep_start();
}

void loop()
{
}
```

Expected Result



Example 2: get NTP time and display on e-ink screen

Get the current time from NTP server and display on the e-ink screen.

Note: please revise ssid and password when uploading codes.

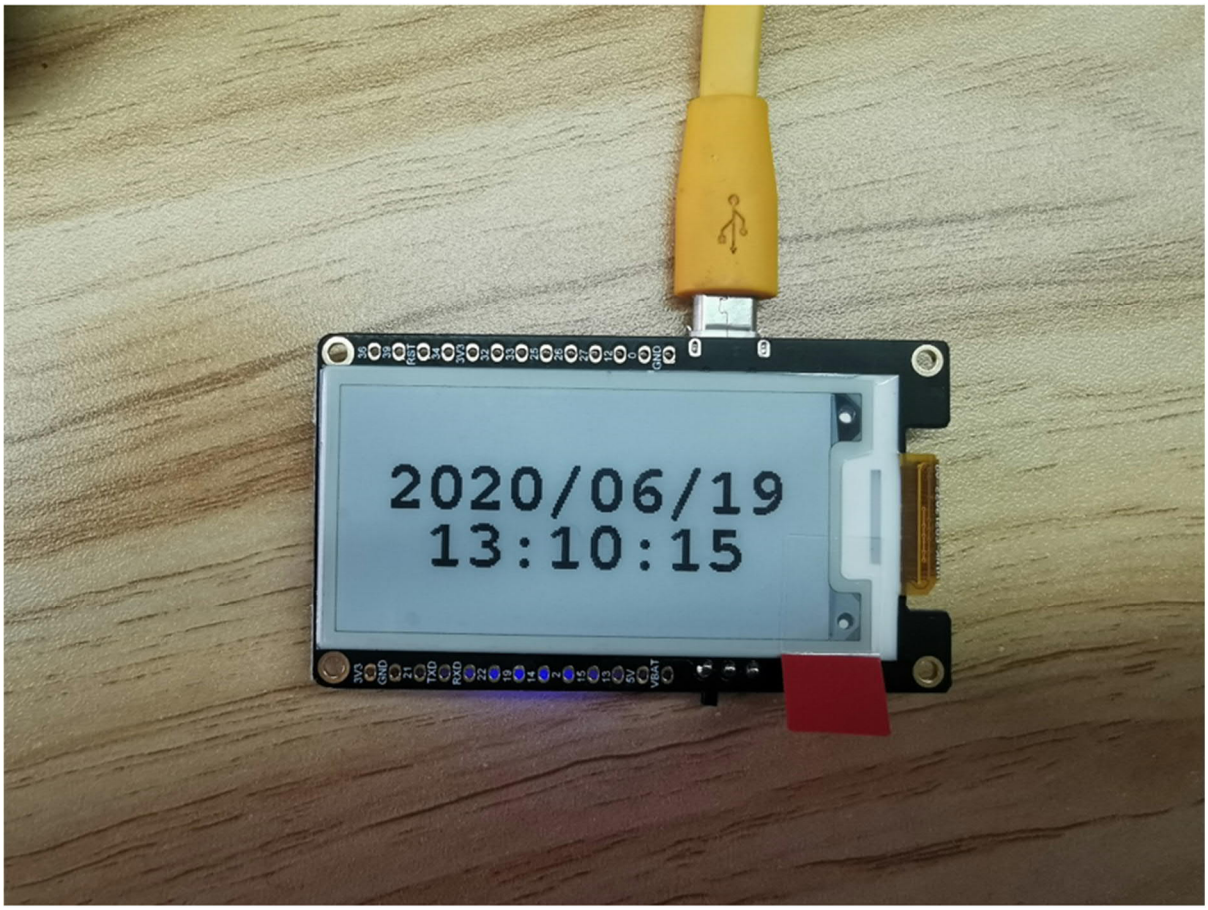
```
#include <GxEPD.h>
#include "SD.h"
#include "SPI.h"


#include <WiFi.h>
#include "time.h"

const char* ssid       = "*****";//wlan information
const char* password   = "*****";
```

```
const char* ntpServer = "ntp.ntsc.ac.cn";//local ntp server
const uint32_t  gmtOffset_sec = 8*3600;  //GMT+08:00
const uint16_t   daylightOffset_sec = 0;


//Since there are multiple versions of the screen, if there is a flower screen
after downloading the program, please test the following four header files again!
#include <GxDEPG0213BN/GxDEPG0213BN.h>
//#include <GxGDE0213B1/GxGDE0213B1.h>      // 2.13" b/w
//#include <GxGDEH0213B72/GxGDEH0213B72.h>  // 2.13" b/w new panel
//#include <GxGDEH0213B73/GxGDEH0213B73.h>  // 2.13" b/w newer panel


#include <Fonts/FreeMonoBold18pt7b.h>


#include <GxIO/GxIO_SPI/GxIO_SPI.h>
#include <GxIO/GxIO.h>

#define SPI_MOSI 23
#define SPI_MISO -1
#define SPI_CLK 18

#define ELINK_SS 5
#define ELINK_BUSY 4
#define ELINK_RESET 16
#define ELINK_DC 17

#define SDCARD_SS 13
#define SDCARD_CLK 14
#define SDCARD_MOSI 15
#define SDCARD_MISO 2

#define BUTTON_PIN 39

typedef enum
{
    RIGHT_ALIGNMENT = 0,
    LEFT_ALIGNMENT,
    CENTER_ALIGNMENT,
} Text_alignment;



GxIO_Class io(SPI, /*CS=5*/ ELINK_SS, /*DC=*/ ELINK_DC, /*RST=*/ ELINK_RESET);
GxEPD_Class display(io, /*RST=*/ ELINK_RESET, /*BUSY=*/ ELINK_BUSY);

SPIClass sdSPI(VSPI);

const uint8_t Whiteboard[1700] = {0x00};

uint16_t Year = 0 , Month = 0 , Day = 0 , Hour = 0 , Minute = 0 , Second = 0;
char Date[]={"2000/01/01"};
char Time[]={"00:00:00"};
bool sdOK = false;

void displayText(const String &str, uint16_t y, uint8_t alignment)
{
```

```
  int16_t x = 0;
  int16_t x1, y1;
  uint16_t w, h;
  display.setCursor(x, y);
  display.getTextBounds(str, x, y, &x1, &y1, &w, &h);

  switch (alignment)
  {
  case RIGHT_ALIGNMENT:
    display.setCursor(display.width() - w - x1, y);
    break;
  case LEFT_ALIGNMENT:
    display.setCursor(0, y);
    break;
  case CENTER_ALIGNMENT:
    display.setCursor(display.width() / 2 - ((w + x1) / 2), y);
    break;
  default:
    break;
  }
  display.println(str);
}

void getTimeFromNTP()
{
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
  }

  Date[2] = (timeinfo.tm_year - 100) / 10 % 10 + '0';
  Date[3] = (timeinfo.tm_year - 100) % 10 + '0';
  Date[5] = (timeinfo.tm_mon + 1) / 10 % 10 + '0';
  Date[6] = (timeinfo.tm_mon + 1) % 10 + '0';
  Date[8] = timeinfo.tm_mday / 10 % 10 + '0';
  Date[9] = timeinfo.tm_mday % 10 + '0';

  Time[0] = timeinfo.tm_hour / 10 % 10 + '0';
  Time[1] = timeinfo.tm_hour % 10 + '0';
  Time[3] = timeinfo.tm_min / 10 % 10 + '0';
  Time[4] = timeinfo.tm_min % 10 + '0';
  Time[6] = timeinfo.tm_sec / 10 % 10 + '0';
  Time[7] = timeinfo.tm_sec % 10 + '0';

  Serial.println(Date);
  Serial.println(Time);
  Serial.println(" ");
}

void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.println("setup");

  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
```

```
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" CONNECTED");

  SPI.begin(SPI_CLK, SPI_MISO, SPI_MOSI, ELINK_SS);
  display.init(); // enable diagnostic output on Serial

  display.setRotation(1);
  display.fillScreen(GxEPD_WHITE);
  display.setTextColor(GxEPD_BLACK);
  display.setFont(&FreeMonoBold18pt7b);
  display.setCursor(0, 0);

  sdSPI.begin(SDCARD_CLK, SDCARD_MISO, SDCARD_MOSI, SDCARD_SS);

  if (!SD.begin(SDCARD_SS, sdSPI)) {
    sdOK = false;
  } else {
    sdOK = true;
  }

  display.fillScreen(GxEPD_WHITE);
  display.update();
}

void loop()
{
  getTimeFromNTP();
  displayText(String(Date), 60, CENTER_ALIGNMENT);
  displayText(String(Time), 90, CENTER_ALIGNMENT);
  display.updateWindow(22, 30,  222,  90, true);
  display.drawBitmap(Whiteboard, 22, 31,  208, 60, GxEPD_BLACK);
}
```

Expected Results

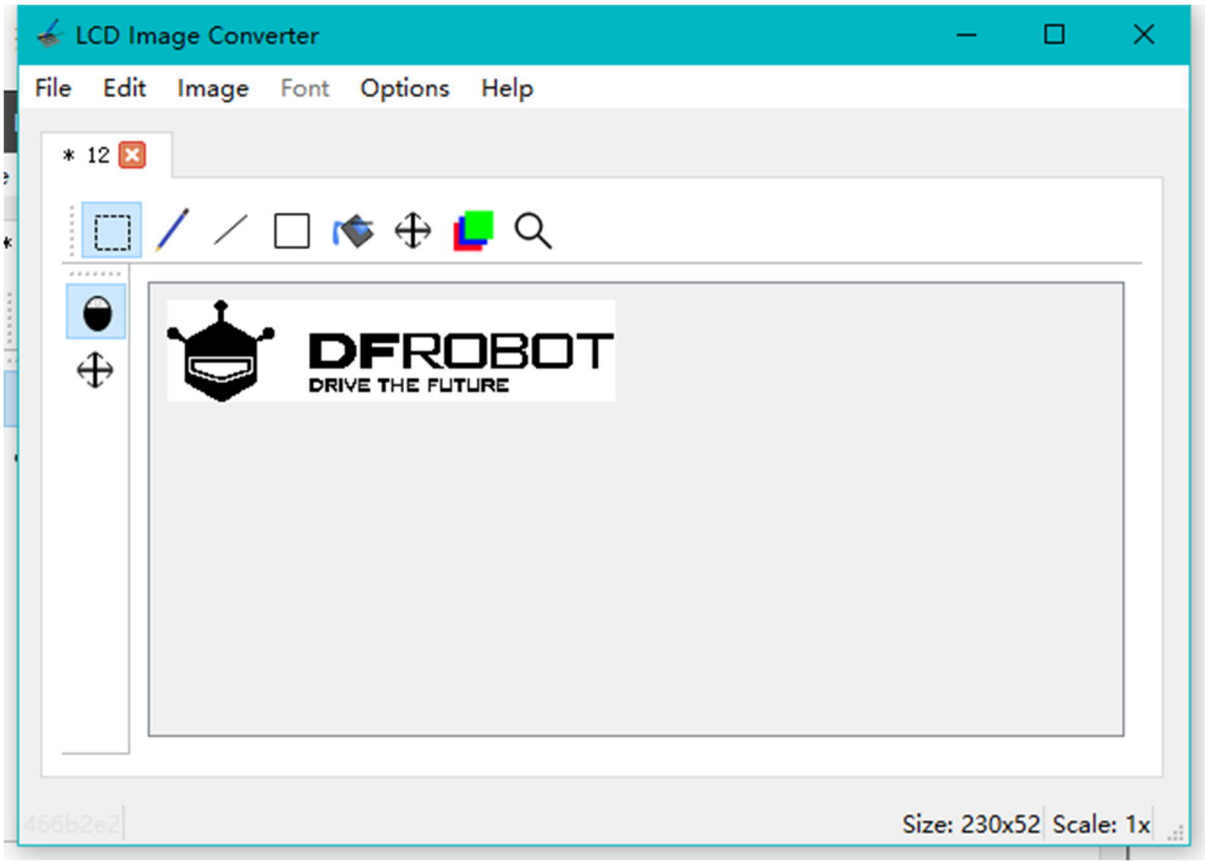


Imagematrix Tutorial

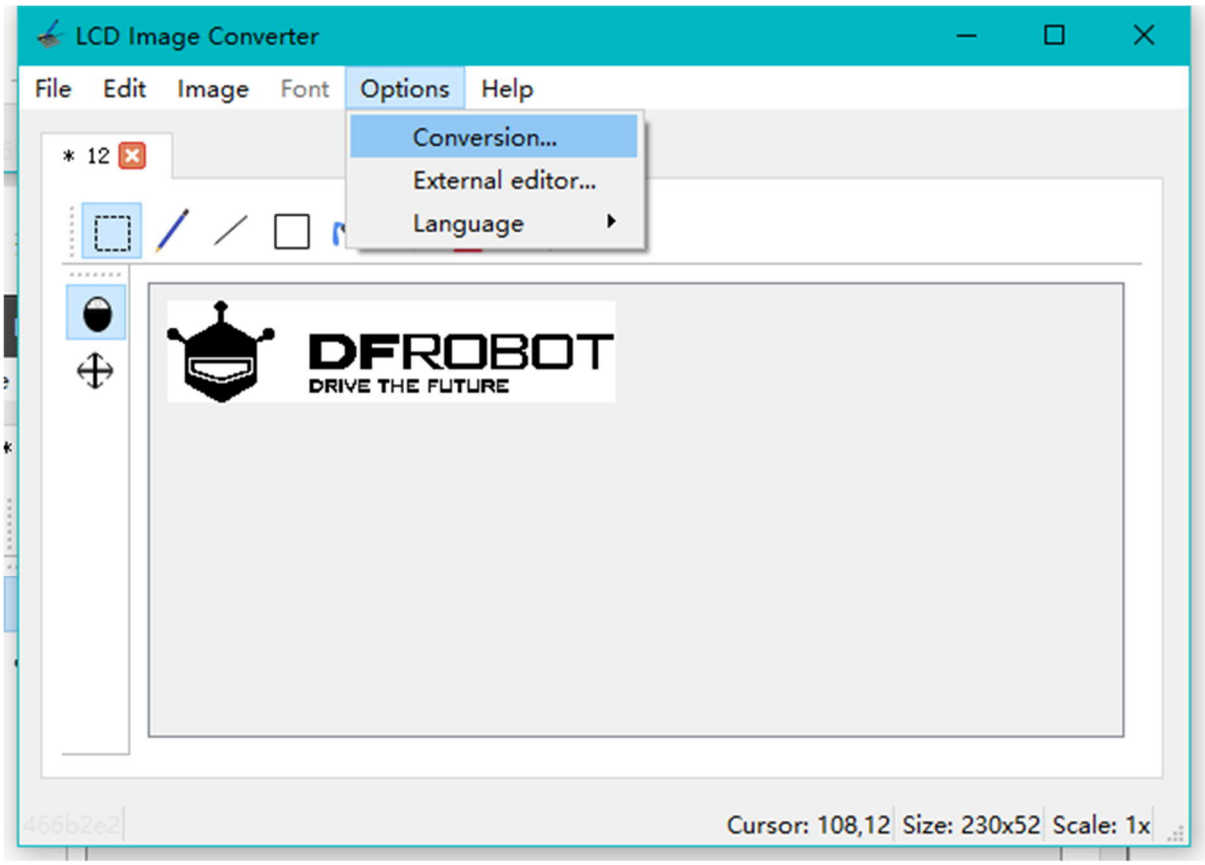
you need to make pure black-and-white pictures, and saved as.bmp or jpg files.

Imagematrix Method The [lcd-image-converter.exe]() software can be used in the imagematrix.

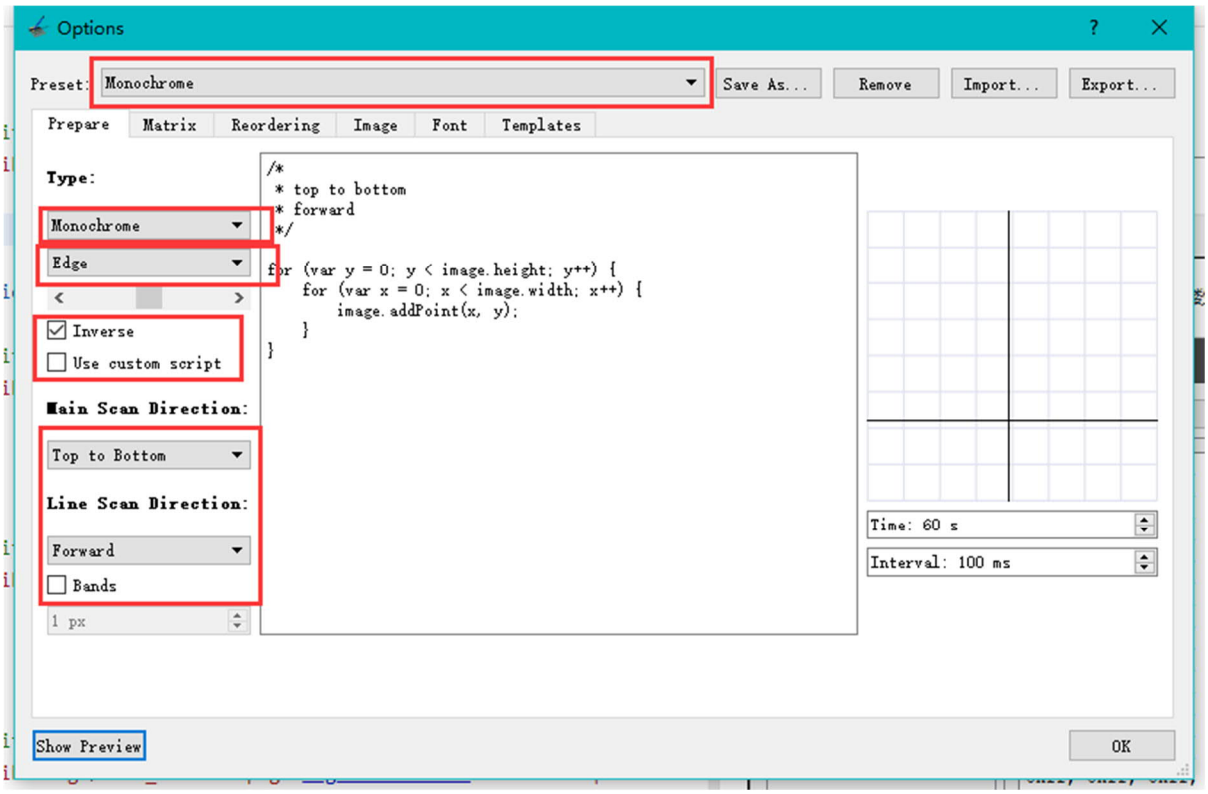
- Open a picture "xxx.jpg". File->open



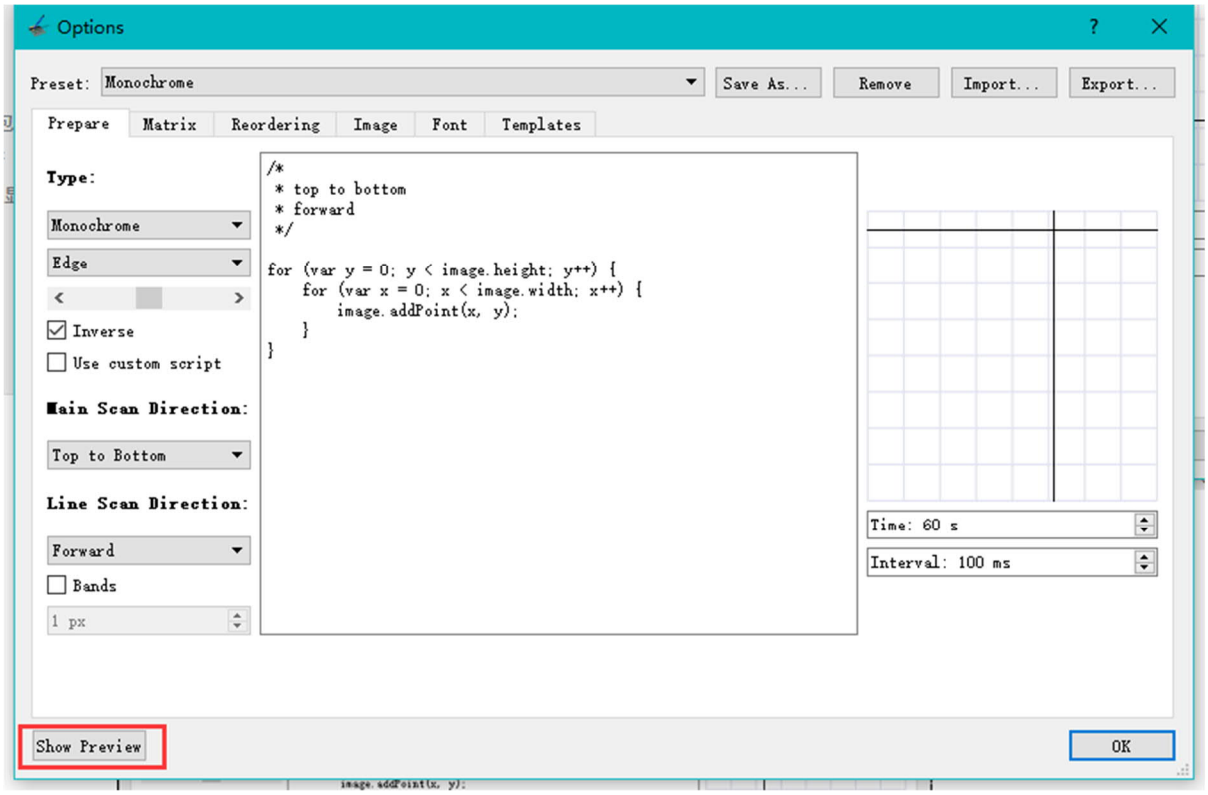
- Open the menu to set the output format. Options->conversion



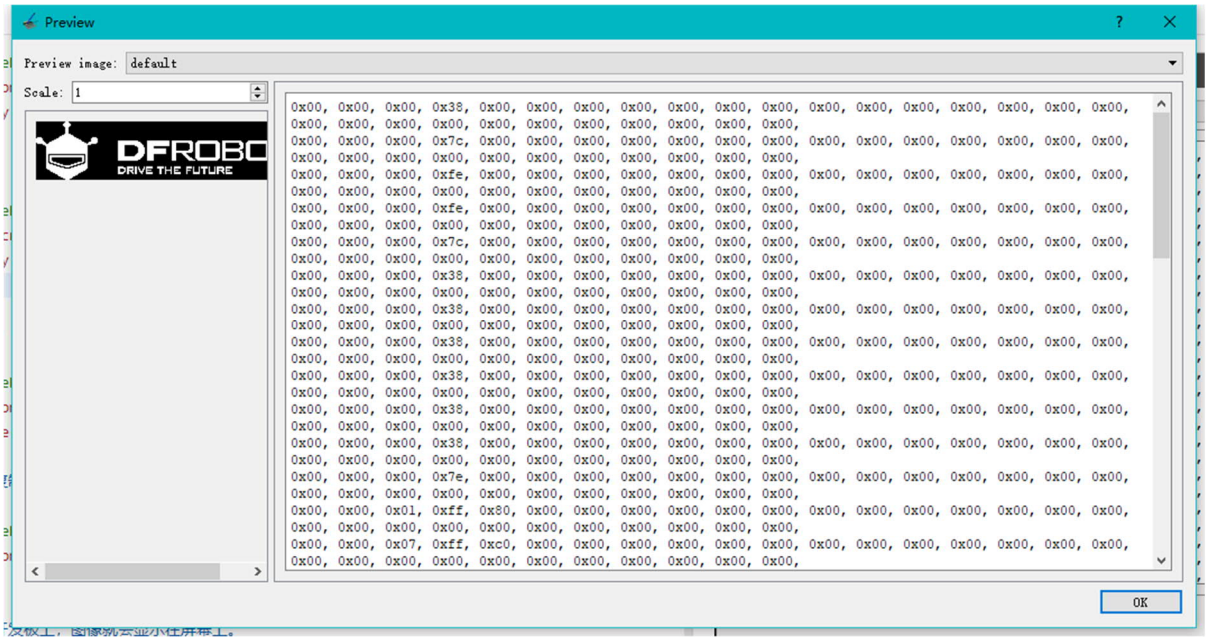
- Set the parameters according to the following figure.



- Click "Show Preview" after setting the parameters.



- Replaces the array in the example with binaries in the dialog box.



- Then, download the program to mainboard and the image will be displayed on the screen